Learned Adaptive Nonlinear Filtering for Anisotropic Diffusion Approximation in Image Processing. *

Bruce Fischl †
†Dept. Cognitive and Neural Systems
e-mail: fischl@cns.bu.edu

Eric L. Schwartz ‡
‡Dept. Cognitive and Neural Systems
e-mail: eric@thing4.bu.edu

Abstract

In the machine vision community multi-scale image enhancement and analysis has frequently been accomplished using a diffusion or equivalent process. Linear diffusion can be replaced by convolution with Gaussian kernels, as the Gaussian is the Green's function of such a system [9]. In this paper we present a technique which obtains an approximate solution to a non-linear diffusion process via the solution of an integral equation which is the nonlinear analog of convolution. The kernel function of the integral equation plays the same role that a Green's function does for a linear PDE, allowing the direct solution of the nonlinear PDE for a specific time without requiring integration through intermediate times. We then use a learning technique to approximate the kernel function for arbitrary input images. The result is an improvement in speed and noise-sensitivity, as well as providing a means to parallelize an otherwise serial algorithm.

1 Introduction.

Multi-scale image enhancement and representation is an important part of biological and machine early vision systems. The process of constructing this representation must be both rapid and insensitive to noise, while retaining image structure at all scales. This is a complex task as small scale structure is difficult to distinguish from noise, while larger scale structure requires more computational effort. In both cases good localization can be problematic. Errors can also arise when conflicting results at different scales require cross-scale arbitration.

Biological systems solve these problems in parallel through the use of retinal and cortical neurons sensitive to stimuli at a variety of spatial scales. Attempts to address issues of this type in the machine vision community resulted in the scale-space formulation of Witken [19] in which an image is convolved with Gaussian kernels of various sizes. Koenderink [9] and Hummel [8] have pointed out that the one-parameter family of images comprising scale-space can be equivalently viewed as snapshots of the time-evolution of the diffusion equation:

$$I_t = c\Delta I \tag{1}$$

Where I is the intensity image, c is a diffusion constant, I_t is the partial derivative of I with respect to time, and Δ is the Laplacian operator with respect to the spatial coordinates. The solution to equation (1) on an infinite domain can be written in terms of the Green's[1] function of the system as

$$I(x,y,t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} I(x',y',0)G(x-x',y-y',t)dx'dy' \tag{2}$$

Where $I(x',y',0)$ is the initial image, and the Green's function G is a Gaussian which is the kernel of the integral operator which is the inverse of the diffusion operator. Thus, convolution with larger scale Gaussian kernels is equivalent to the evolution of the diffusion equation for longer periods of time, with the original image as initial conditions.

The diffusion equation provides a mathematical framework with which to analyze the scale-space formalism, but it does not address the issue of cross-scale comparison. While Koenderink restricted his analysis to the isotropic diffusion characterized by the linear heat equation, Perona and Malik [13, 14] suggested that a nonlinear anisotropic version of the heat equation could remedy some of the difficulties encountered in the use of a linear scale-space. This was similar to earlier neural modeling work [2] which used context-sensitive diffusion on a discrete neural lattice. Malik and Perona proposed that the conduction coefficient be a function of the intensity gradient magnitude of the image:

$$I_t = \nabla \cdot (c(\mid \nabla I \mid)\nabla I) \tag{3}$$

*Supported in part by the office of naval research (ONR N00014-95-1-0409).

[1] More accurately, the Green's function is the Gaussian multiplied by a temporal step function [1]

1015-4651/96 $5.00 © 1996 IEEE
Proceedings of ICPR '96

Making $c(\cdot)$ a decreasing function of intensity gradient magnitude causes regions of high contrast to undergo less diffusion, and be preserved over time. This is in contrast to the linear heat equation which blurs uniformly, destroying small scale structure as time evolves. Systems such as equation (3) are intended to yield a single intensity image which retains edge information at all scales of interest, thus obviating the need for any type of cross-scale arbitration.

The Perona-Malik equation (3) is a nonlinear partial differential equation of a type which is difficult to analyze. It has been suggested [10] that (3) is unstable for some parameter regimes, although this is still a point of investigation [15]. Furthermore, it can amplify small scale noise which gives rise to high gradient magnitudes. Many variants of the Perona and Malik scheme have been proposed to improve its sensitivity to noise, its instability, and its equilibrium behavior (see [16] for a review).

The diffusion paradigm, while impressive in the quality of the images it produces, suffers from a number of drawbacks. The most prominent of these is the computational cost of the algorithms, coupled with their inherently serial nature. This makes them implausible from a biological standpoint, as well as impractical for use in real-time machine vision applications. The biological implausibility stems from the relatively rapid nature of perception relative to neural conduction delays and peak firing rates($\leq 200Hz$). Psychophysical and neurophysiological experiments indicate that perception can occur as rapidly as 60-100 msec [17, 12] which is comparable to the latency of cells in V1 [11]. Using this figure, Thorpe and Imbert [17] argue that the number of synaptic connections used by the visual system in rapid identification tasks is somewhere between 10 and 50, although probably closer to the lower bound. Thus, while complex processing is possible, it is almost certainly parallel in nature.

In this paper we propose a method to directly learn the input-output mapping of the diffusion process. This has a number of advantages. Most importantly it parallelizes the inherently serial process of numerical integration. In addition, it obviates the need for regularization of (3) to preserve image structure at equilibrium, as an appropriate time constant is implicitly imbedded in the system. Further, it is an order of magnitude faster than the full diffusion process, even when run on serial machines. The end result is an approximation of the kernel function in equation (2) for arbitrary input images. While we present the algorithm in the context of anisotropic diffusion, the technique is applicable to a much wider range of image enhancements/modifications.

2 Construction of an Integral Solution.

The advantage of the use of a Green's function in the solution of the linear heat equation is twofold. First, the Green's function generates a solution for the desired time without requiring the traversal of solutions at intervening times. Second, the integral form of the solution is amenable to parallel implementation. In order to obtain both of these advantages for the nonlinear heat equation (3) we take a similar approach and attempt to find the kernel function of the integral operator such that

$$I(x,y,t) = \int\int I(x',y',0)G_t(x,x',y,y')dx'dy' \quad (4)$$

Where the kernel function is subscripted with with t to emphasize the fact that different kernel functions exist for different evolution times. Although equation (4) is ill-posed, we use the following simple idea to calculate the appropriate kernel function. At each iteration of the numerical integration of the PDE (3), we track the diffusion path of the intensity values. When carried out for n time steps, this allows the iterative computation of $G_{n\Delta t}$.

The function G_t of equation (4) consists of an array of two dimensional kernels, which we term diffusion kernels, one for each point in the image. We denote the kernel at image point (x,y) by $C_{x,y}$. The value of this kernel at image location $(x+i, y+j)$ is $C_{x,y}(i,j)$. The construction of the kernels is dependent on the form of the numerical integration of the diffusion equation. The algorithm we employ (see [4]) obtains the image at time $t+\Delta t$ via correlation of the image at time t with a set of space and time varying masks:

$$I(x,y,t+\Delta t) = \sum_{x'}\sum_{y'} K^t_{x,y}(x',y')I(x+x',y+y',t) \quad (5)$$

With the mask weights given by:

$$K^t_{x,y} = \frac{\Delta t}{2}\begin{bmatrix} 0 & c^N(t) & 0 \\ c^W(t) & \frac{2}{\Delta t} - (\sum_{i\neq 0} c^i(t)) & c^E(t) \\ 0 & c^S(t) & 0 \end{bmatrix} \quad (6)$$

Where the superscripts 0,N,E,S,W refer to the central pixel and its four connected neighbors, and $\Delta t \leq 0.25$ for the algorithm to be numerically stable [7]. Thus at time t, the value of the mask associated with the image point (x,y) at location $(x+i, y+j)$ is given by $K^t_{x,y}(i,j)$. Since the four dimensional nature of these objects causes unnecessary notational clutter, we will proceed in one spatial dimension. The generalization to two dimensions is straightforward.

Before commencing with the details of the algorithm it is worth commenting on the different roles the masks and kernels play. The mask $K^t_x(i)$ relates the image value at time t at position $x+i$ to the image value at time $t+1$ at

position x.

$$I(x,t+1) = \sum_i K_x^t(i) I(x+i,t) \quad (7)$$

Conversely, the kernel entry $C_x^t(j)$ prescribes the contribution of the initial image value at position $x+j$ to the image value at position x at time t

$$I(x,t) = \sum_j C_x^t(j) I(x+j,0) \quad (8)$$

Equation (8) is a discrete statement of the Green's function property of the kernels, while equation (7) reiterates the role of the masks in the numerical integration of the PDE.

In order to construct the kernels $C_x^t(i)$ we proceed inductively. For each point x in the image, we create a kernel and initialize it using a Kronecker delta function, $C_x^0(i) = \delta_i$. The application of this set of kernels to the initial image leaves it unchanged, and therefore equation (8) hold for $t = 0$. We obtain a recursive update rule for constructing the diffusion kernel at time $t+1$ given the kernels at time t by combining equations (8) and (7). This yields the relationship between the initial image and the image at time $t+1$:

$$I(x,t+1) = \sum_i K_x^t(i) \sum_j C_{x+i}^t(j) I(x+i+j,0) \quad (9)$$

Since we are seeking $C_x^{t+1}(n)$, the kernel element at some arbitrary position n, we are only interested in the coefficients of $I(x+n,0)$ in (9). Examining (9) for values of j such that $i+j=n$, we arrive at the recursive update law:

$$C_x^{t+1}(n) = \sum_i K_x^t(i) C_{x+i}^t(n-i) \quad (10)$$

Equation (10) can be understood in the following way. The kernel value $C_{x+i}^t(n-i)$ represents the contribution of the initial intensity value at pixel $x+n$ to the pixel at $x+i$ at time t. $K_x^t(i)$ then gives the proportion of the total intensity at location x at time $t+1$ which comes from position $x+i$. By summing over all i, we account for all possible paths the intensity value $I(x+n,0)$ can diffuse through and arrive at position x at time $t+1$.

In practice, we must limit the support of the diffusion kernels, which requires one additional modification. The value $C_x(j)$ is the percentage contribution of the initial image intensity at $x+j$ to the final image value at x, and therefore must sum to one. If the support of the C_x is the entire image, this occurs naturally. However, in order to conserve image intensity when a limited support is imposed, each C_x must be explicitly normalized after every time step.

3 Kernel estimation.

Given the kernel function construction algorithm described in section 2, we now seek a means of approximating an appropriate kernel function for an arbitrary image. That is, given a novel image, we want to estimate a set of kernels which yield an image that is perceptually similar to the one that would have been obtained by integrating the diffusion equation.

Even limiting the support of the kernels, the output of the kernel estimation process is relatively high dimensional. Since the amount of data required to fit an n-dimensional function increases exponentially with the dimensionality, we must compress the kernel representation considerably. To accomplish this, we perform a principal components analysis (PCA) of the smoothed kernels which yields a set of basis functions for kernel construction, as well as the coefficients of the expansion, which must be estimated for each specific image.

The estimation of the PCA coefficients is accomplished using a multi-layer perceptron as a function approximator trained with a modified backpropagation algorithm [18]. The function approximation task is simplified somewhat due to the uncorrelated nature of the principal components. This allows us to train separate networks for each component, obviating the need to learn nonlinear interactions between coefficients.

The features used as inputs to the function approximator must have some degree of noise tolerance while also representing the image structure in a neighborhood of each point. A simple scheme which satisfies these criteria is the use of a 3x3 window of Gaussian smoothed intensity gradient magnitude based on x and y image derivatives generated by a Sobel operator. The 3x3 neighborhood gives the network coarse information about both local orientation (if one exists), as well as edge offset.

4 Results.

We have applied the Green's function approximation or GFA algorithm to two different diffusion schemes. The filter used in most of this paper was constructed by training a multilayer perceptron on coefficients derived from a single 64x64 pixel image (not shown in this paper). The kernels were constructed by iterating the Malik-Perona diffusion algorithm on the training image for 100 time steps except where noted. All images are scaled so that their intensity values are between zero and one, all edge maps are generated using the same parameter set[2], and all diffusion generated images are created using the same number of iterations and identical parameter values ($k = 0.05$ or 50).

[2]We use the Matlab 'edge' routine which employs Sobel operators and thresholded non-maximum suppression to generate edge maps.

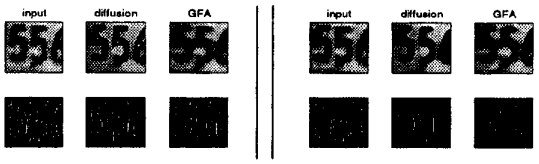

Figure 1. GFA vs. Malik-Perona (left) and mean-curvature (right, $k = 50$) diffusion.

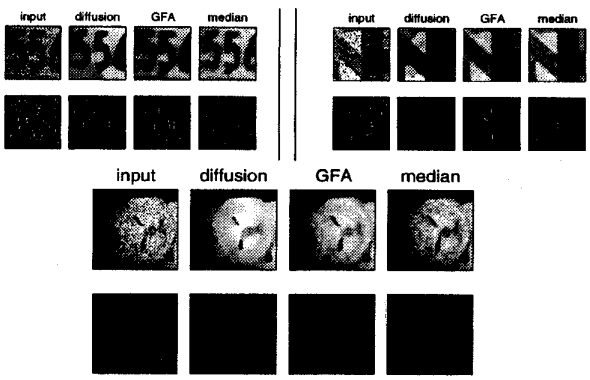

Figure 2. Filter response to Gaussian (left), salt & pepper (right), as well as multiplicative noise (bottom).

4.1 Gradient and mean curvature-based diffusion.

The two diffusion schemes we approximate as examples of the GFA filter are the gradient-based diffusion of Perona and Malik [13, 14] which uses a conductance function such as c_1 in equation (11), and the mean curvature-based diffusion of El-Fallah and Ford [3] which employs c_2 as a conductance function

$$c_1(|\nabla I|) = e^{-\left(\frac{|\nabla I|}{k}\right)^2}, c_2(|\nabla I|) = \frac{1}{\sqrt{1 + k^2|\nabla I|^2}} \quad (11)$$

Figure (1) left displays the results of applying the Malik-Perona equation as well as the GFA filter to a novel real-world image, corrupted with naturally occurring noise. The noise-enhancing property of the Malik-Perona process is apparent, as speckle noise in all three digits has been retained and expanded into white blocks in each character. The GFA filtered image does not evidence this effect, as the interior of all three characters is uniform.

El-Fallah and Ford [3] proposed their conductance function as a remedy for the lack of noise tolerance in the Malik and Perona approach. Figure (1) right depicts the mean curvature based diffusion as well as its approximation. As can be seen, the mean curvature-based diffusion has excellent noise suppression qualities. The character outlines are quite clear, with only minor noise edges remaining. The GFA filter used to create the middle image was generated by using kernels from the curvature-based diffusion process, and has properties similar to the filter used elsewhere in this paper.

4.2 Noise tolerance and comparison with median filtering.

While many filters are capable of reconstructing images for a given (known) noise distribution, the restoration problem is much more difficult if the noise characteristics are unknown or variable. The GFA filter noise-tolerance properties derive from a number of its components. The Gaussian blurred gradient magnitude, coupled with the neural network, combine to form a robust function approximator in the presence of a wide assortment of noise profiles. In this section we compare the GFA filter to both the mean curvature diffusion process as well as a 5x5 median filter on noise corrupted images. The median filter is a simple image processing technique useful for some types of noise-reduction. The first noise model we use to illustrate the performance of the GFA filter is an additive Gaussian one. Figure (2) left contrasts the GFA performance with that of a median filter applied to an image corrupted with zero mean, 0.25 variance additive Gaussian noise. The edge maps at the bottom indicate that the GFA filtered image contains most of the edge information of the image generated by applying the mean-curvature diffusion model at far less cost, while the median filter breaks down at this noise level.

Figure (2) bottom illustrates the excellent performance of the GFA filter in the presence of multiplicative noise (10% amplitude) as compared to the median filter. In this case, the full diffusion process washes away all detail of the fur and fuses the nose and mouth into a single region, while the GFA filter eliminates most of the noise but retains much of the fur texture as well as the nose and mouth borders.

As the last noise model, we demonstrate the GFA filter performance in the presence of salt and pepper noise (random pixel values replaced with zeros or ones, 10% density). In this type of environment the median filter produces excellent quality images, as evidenced by the building shown in figure (2) right. This is due to the invariance of the median filter to noise amplitude as opposed to noise density which plagued it in earlier examples. The mean curvature-based diffusion, which performs so well in the presence of other types of noise, encounters difficulty with salt and pepper noise. As can be seen, region boundaries for the triangular patches on the left side of the building have been lost, and new borders have been generated which do not correspond to any real image feature. In contrast, the GFA filter retains most of the image structure, as is reflected in the edge maps

at the bottom.

5 Conclusion.

Diffusion is a powerful tool of great potential utility in early vision. It unifies multi-scale processing into a simple procedure which reduces noise and integrates information at all scales of interest. However, it is a computationally costly and inherently serial process. The GFA algorithm reduces the computational cost of diffusion as well as transforming the process into one that is amenable to parallelization. It also resolves many of the issues that are problematic for the Perona and Malik approach. Regularization to improve equilibrium behavior is unnecessary as an appropriate time constant is implicitly imbedded in the system. Noise sensitivity can be dealt with separately in the feature extraction and training stages, and is thus no longer an issue for the PDE. Potential numerical instability of the underlying PDE is also eliminated as images for which the instability is manifest can be excluded from the training set.

From a biological standpoint, the GFA type approach is well suited for use in the mammalian retino-cortical system. Feature extraction via convolution as well as basis function expansion are reasonable interpretations of neural receptive fields. The massive parallelism present in these systems would allow the use of a large number of basis functions, input features, as well as units for learning and approximation, without sacrificing computational speed as on serial machines. This should dramatically increase the noise tolerance and accuracy of the process.

While many image enhancement processes yield impressive machine vision as well as psychophysical modeling results [5, 6] many of them suffer from the same drawbacks as the anisotropic diffusion paradigm: inherent serialism coupled with costly integration times. The methods outlined in this paper can be applied to these techniques, retaining many of the benefits of the underlying procedure while improving their utility.

In summary, we have presented a novel filter which learns an approximate Green's function from examples of a nonlinear diffusion process. The learned filter can then directly generate output images which are good approximations of the diffusion. The technique is applicable to image processing algorithms other than anisotropic diffusion. The GFA filter is tolerant of a wide variety of noise environments, parameter free, robust, and between one and two orders of magnitude faster than the full-scale anisotropic diffusion on a serial architecture, and is amenable to full parallelization.

References

[1] G. Barton. *Elements of Greens Functions*. Oxford University, 1991.

[2] M. A. Cohen and S. Grossberg. Neural dynamics of brightness perception: Features, boundaries, diffusion, and resonance. *Perception and Psychophysics*, 36:428–456, 1984.

[3] A. I. El-Fallah and G. E. Ford. Nonlinear adaptive image filtering based on inhomogeneous diffusion and differential geometry. *SPIE Image and Video Processing II*, 2182:49–63, 1994.

[4] B. Fischl and E. Schwartz. Learning an integral equation approximation to nonlinear anisotropic diffusion in image processing. Technical Report CAS/CNS-TR-95-033, Boston University, Department of Cognitive and Neural Systems, 677 Beacon St, Boston MA, Dec. 1995.

[5] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6:721–741, 1984.

[6] S. Grossberg and E. Mingolla. Neural dynamics of form perception: Boundary completion, illusory figures, and neon color spreading. *Psychological Review*, 92(2):173–211, 1985.

[7] R. Haberman. *Elementary applied partial differential equations*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, second edition, 1987.

[8] A. Hummel. Representations based on zero-crossings in scale-space. In M. Fischler and O. Firschein., editors, *Readings in Computer Vision: Issues, Problems, Principles and Paradigms*. Morgan Kaufmann, Los Angeles, 1986.

[9] J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[10] M. Nitzberg and T. Shiota. Nonlinear image filtering with edge and corner enhancement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(8):826–833, 1992.

[11] L. G. Nowak, M. Munk, P. Girard, and J. Bullier. Visual latencies in areas v1 and v2 of the macaque monkey. *Visual Neuroscience*, 12:371–384, 1995.

[12] M. W. Oram and D. I. Perrett. Time course of neural responses discriminating different views of the face and head. *Journal of Neurophysiology*, 68(1):70–84, July 1992.

[13] P. Perona and J. Malik. Scale space and edge detection using anisotropic diffusion. In *Proceedings of the IEEE Computer Society Workshop on Computer Vision*, pages 16–27, Miami, FL., 1987.

[14] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[15] P. Perona, T. Shiota, and J. Malik. Anisotropic diffusion. In B. M. T. H. Romeny, editor, *Geometry Driven Diffusion in Computer Vision*, chapter 3, pages 73–92. Kluwer, 1994.

[16] B. M. T. H. Romeny. *Geometry-Driven Diffusion in Computer Vision*. Kluwer, 1994.

[17] S. J. Thorpe and M. Imbert. Biological constraints on connectionist models. In R. Pfeifer, Z. Schreter, and F. Fogelman-Soulie, editors, *Connectionism in perspective*, pages 63–92. Elsevier, Amsterdam, 1989.

[18] P. J. Werbos. *Beyond regression: new tools for prediction and analysis in the behavioral sciences*. PhD thesis, Harvard University, 1974.

[19] A. Witken. Scale-space filtering. In *International Joint Conference on Artificial Intelligence*, pages 1019–1021, 1983. Karlsruhe, West Germany.